

WHAT IS CLAIMED IS:

1. A system for extending interactivity of a presentation markup language, the system comprising:

- 5 a collection of designated elements, each designated element comprising:
 a namespace; and
 attributes for describing features of the designated element; and
 a collection of associated instructions for performing functions to elements in the
document object model, the instructions associated with the designated elements.

10

2. The system as claimed in claim 1, further comprising an initialization function for directing the processing one or more designated elements in the document object model, having instructions for traversing each node in the document object model and for searching and calling functions associated with designated elements having names
15 following a predetermined naming convention.

3. The system as claimed in claim 1, wherein the designated element is associated with an extensible markup language element.

20

4. The system as claimed in claim 1, further comprising:

- a collection of designated attributes applied to one or more of the document object
model elements for applying passive behavior to objects in the web application; and
 a collection of associated instructions for performing functions associated with the
designated attributes.

25

5. The system as claimed in claim 4, wherein the collection of designated attributes comprises one or more of:

 a 'drag' attribute for specifying whether the element is movable by clicking and
dragging it with the mouse;

30

 a 'pan' attribute for specifying whether the element is immune to panning;

a 'zoom' attribute for specifying whether the element is immune to zooming;
a 'selected' attribute for specifying whether the element has been selected; and
a 'selectionGroup' attribute for specifying an 'id' attribute of a <selection>
element that this element is associated with.

5

6. The system as claimed in claim 1, wherein the collection of designated elements
comprises one or more of:

flow control elements for controlling statement flow of the web application;
coordinate mapping elements for manipulating coordinates of objects in the web

10 application;

behavior elements for manipulating viewer behavior with respect to the web
application;

a focus element for selecting a group of elements in the web application; and

a constraint element for constraining manipulable attributes of an element in a

15 web application.

7. The system as claimed in claim 6, wherein the flow control elements comprise one or
more of:

an if element for defining a simple conditional statement which, when it evaluates
20 to true, results in its child elements being executed or rendered;

a switch element for defining a conditional statement, and for comparing one
value to other values defined in child <case> elements;

a case element for defining the value to compare to a 'value' attribute of the
switch element;

25 a default element for containing action elements to be executed; and

a loop element for defining a repeated sequence of actions.

8. The system as claimed in claim 6, wherein the coordinate mapping elements comprise
one or more of:

a mousePosition element for defining a container for holding current mouse coordinates;

a mapCoords element for defining an object used for mapping coordinates in one space to another space, via a polynomial transformation, whose coefficients are

5 determined by the coordinates of point-pairs;

a pointPair element for defining x-y coordinates for a same location in two different coordinate spaces; and

a mapProj element for defining an object used for mapping coordinates in one projection system to another.

10

9. The system as claimed in claim 6, wherein the viewer behavior elements comprise one or more of:

a zoom element for scaling a document by a factor;

a pan element for translating a document by an amount; and

15 a playSound element for playing an audio file.

10. A method of extending interactivity of presentation markup languages, the method comprising one or more of the following:

controlling statement flow of a web application, the method comprising the steps

20 of:

searching for a flow control element in a document object model of the web application;

generating a function name associated with the flow control element;

calling the generated function name; and

25 processing child elements of the flow control element;

coordinate mapping of a web application, the method comprising the steps of:

searching for a coordinate mapping element in a document object model of the web application;

generating a function name associated with the coordinate mapping

30 element; and

calling the generated function name;
manipulating viewer behavior with respect to a web application, the method
comprising the steps of:
searching for a viewer behavior element in a document object model of the
web application;
generating a function name associated with the viewer behavior element;
and

calling the generated function name;
focussing a group of elements in a web application, the method comprising the
steps of:

searching for a focus element in a document object model of the web
application;
generating a function name associated with the focus element; and
calling the generated function name;

constraining manipulable attributes of an element in a web application, the method
comprising the steps of:

searching for a constraint element in a document object model of the web
application;
generating a function name associated with the constraint element; and
calling the generated function name; and

applying passive behavior to an element of a web application, the method
comprising the steps of:

searching for a designated attribute of the element in a document object
model of the web application;
generating a function name associated with the designated attribute; and
calling the generated function name.

11. A method of extending interactivity of a presentation markup language, the method
comprising the steps of:

searching for a designated control element in a document object model; and

calling a function associated with the designated control element.

12. The method as claimed in claim 11, wherein the step of searching includes the steps of:

5 traversing each node in the document object model; and
 determining whether an element has a name which follows a designated naming convention.

13. The method as claimed in claim 11, wherein the step of calling a function includes
10 the steps of:

 dynamically generating a function name associated with the designated element;
 passing an object associated with the designated element as a parameter of the
 generated function;
 retrieving the attributes of the object; and
15 performing a function stored in memory having the generated function name.

14. The method as claimed in claim 13, wherein the step of dynamically generating
includes the steps of:

20 determining if the name of the designated element contains a designated prefix;
 generating a function name comprising of the name of the designated element;
 assigning an object associated with the designated element as the parameter of the
 function; and
 assigning predetermined instructions of the designated element as steps for the
 function to perform.

25

15. The method as claimed in claim 11, wherein the step of calling a function includes
the steps of:

 determining which script in a collection of scripts is associated with the
 designated element; and
30 calling the script.

16. The method as claimed in claim 11, further comprising the steps of:
searching for a designated attribute in an element in a document object model; and
calling a script associated with the designated attribute.

5 17. The method as claimed in claim 16, wherein the step of searching for a designated attribute comprises the steps of:

searching attributes of an element in a document object model;
determining whether an element attribute has a name which follows a designated
naming convention.

10

18. The method as claimed in claim 16, wherein the step of calling a script includes the steps of:

determining if the name of the designated attribute contains a designated prefix;
generating a function name comprising of the name of the designated attribute;
15 assigning an object associated with the designated attribute as the parameter of the function name ; and
assigning predetermined instructions of the designated attribute as steps for a function having the function name to perform.

20 19. The method as claimed in claim 16, wherein the step of calling a script includes the steps of:

dynamically generating a function name associated with the designated attribute;
passing an object associated with the designated attribute as a parameter of the generated function name;
25 receiving the attributes of the object; and
performing a function stored in memory having the generated function name.

20. The method as claimed in claim 19, wherein the step of dynamically generating comprises the steps of:

30 determining if the name of the designated attribute contains a designated prefix;

generating a function name comprising of the name of the designated attribute;
assigning an object associated with the designated attribute as the parameter of the
function; and

5 assigning predetermined instructions of the designated attribute as steps for the
function to perform.

21. The method as claimed in claim 19, wherein the step of calling a script includes the
steps of:

10 determining which script in a collection of scripts is associated with the
designated attribute; and
 calling the script.

22. A method of extending interactivity of a presentation markup language by controlling
user interface features of a web application, the method comprising the steps of:

15 adding a behavior element as a child of a designated element;
 receiving an event which is equal to an event attribute setting in the behavior
element; and
 calling a script associated with the behavior element.